AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A process for reordering items in a database to be retrieved for display to a user, comprising the steps of:

ordering a plurality of words stored in a linguistic database (LDB) according to a predefined linguistics frequency of use model;

associating each word stored in said LDB with an object number;

accepting user input from a keyboard;

retrieving any words from said LDB that match said user's input;

displaying a list of said retrieved words as ordered in said LDB;

if said list does not include a word, accepting said word as defined by said user;

storing said user-defined word in a user database (UDB);

assigning a frequency count to said user-defined word;

storing said frequency count in said UDB;

~~ordering words in a linguistic database (LDB) according to a linguistic frequency of use model;~~

~~ordering words in a user database (UDB) separate from the LDB that stores user defined words that the user enters into the system and that includes a reorder database (RDB);~~

~~storing in the RDS object numbers for words contained in the LDB;~~

~~dynamically retrieving from said LDB and from said UDB words that include the sequence of letters formed by the user input;~~

~~displaying to the user a list of retrieved words, wherein the order of words determined by the RDB are displayed before the order of words determined by the LDB;~~

enabling ~~the~~ said user to select a word from ~~the~~ said displayed list; ~~and~~

assigning said frequency count to every selected word;

storing said frequency count of said selected word and said word's corresponding object number in said UDB;

~~assigning a dynamic reordering frequency count to words selected by the user and inserting the selected words' assigned reordering frequency counts into the RDB~~

accepting a subsequent user input;

retrieving any words from said LDB and any user-defined words from said UDB that match said user's subsequent input;

if more than one word matches said user's subsequent input, dynamically reordering any matching words in said LDB as ordered in said LDB and any matching words in said UDB according to said frequency count; and

displaying a list of said reordered matching words if more than one word matches.

2.-3. (Cancelled)

4. (Currently Amended) The process of Claim 1, wherein said frequency count is assigned to said selected word if said selected word is in a non first order position in said list of said retrieved words or said list of said reordered words and is selected for a first time ~~assigning step inserts into said reorder database a non first ordered word from said list if the user selects the non first ordered word for the first time, and inserts into said reorder database the nonselected first ordered word if it does not already exist in said reorder database~~.

5. (Currently Amended) The process of Claim 4, wherein if ~~the~~ said non first ordered word is selected by ~~the~~ said user a predetermined number of times then ~~the~~ said non first ordered word is displayed in a ~~assigned a higher frequency count than the~~ first ordered ~~word~~ position in said list of reordered matching words.

6. (Currently Amended) The process of Claim 4, wherein all non first ordered words that are assigned said frequency count ~~entered into said reorder database~~ are initially assigned equal ~~reordering~~ frequency counts.

7. (Currently Amended) The process of Claim 1, wherein saida word's ~~reordering~~ frequency count is increased each time ~~the~~ said user selects ~~the~~ said word.

8. (Currently Amended) The process of Claim 1, wherein if ~~the~~ said user selects said word from said list of retrieved words or said list of reordered words ~~a word which~~ that is below a second ordered position, then said selected word is assigned said ~~assigning step assigns a value to the word's reordering~~ frequency count that places ~~the~~ said word in ~~the~~ said second ordered position in said list.

9. (Currently Amended) ~~The process of Claim 1, further comprising the step of:~~ A process for reordering items in a database to be retrieved for display to a user, comprising the steps of:

ordering a plurality of words stored in a linguistic database (LDB) according to a predefined linguistics frequency of use model;

associating each word stored in said LDB with an object number;

accepting user input from a keyboard;

retrieving any words from said LDB that match said user's input;

displaying a list of said retrieved words as ordered in said LDB;

if said list does not include a word, accepting said word as defined by said user;

storing said user-defined word in a user database (UDB);

assigning a frequency count to said user-defined word;

storing said frequency count in said UDB;

enabling said user to select a word from said displayed list;

assigning said frequency count to every selected word;

storing said frequency count of said selected word and said word's corresponding object number in said UDB;

accepting a subsequent user input;

retrieving any words from said LDB and any user-defined words from said UDB that match said user's subsequent input;

if more than one word matches said user's subsequent input, dynamically reordering any matching words in said LDB as ordered in said LDB and any matching words in said UDB according to said frequency count;

displaying a list of said reordered matching words if more than one word matches; and

periodically aging ~~the reordering~~ said frequency counts in said reorder database by reducing ~~the reordering~~ said frequency counts by a predetermined factor.

10. (Currently Amended) The process of Claim 1, further comprising the step of:

periodically checking free space in said ~~reorder database~~ UDB and, if ~~the~~ said free space is less than a predetermined amount, then removing from said ~~reorder database~~ UDB said words that have ~~reordering~~ said frequency counts below a predetermined threshold.

11. (Currently Amended) The process of Claim 10, wherein said removing step removes said user-defined words having ~~reordering~~ said frequency counts below said predetermined threshold after removing other words having ~~reordering~~ said frequency counts below ~~the~~ said predetermined threshold.

12. (Currently Amended) The process of Claim 1, further comprising the step of:

resolving ~~reordering~~ frequency collisions when two words in said list of reordered words are associated with ~~have~~ equal ~~reordering~~ frequency counts by ordering ~~the~~ said word that was most recently selected ~~of the two words~~ first.

13. (Currently Amended) The process of Claim 1, further comprising the step of:

resolving ~~reordering~~ frequency collisions when two words in said list of reordered words are associated with ~~have~~ equal ~~reordering~~ frequency counts by ordering ~~the~~ said word having a higher initial ordering in said ~~linguistic database~~ LDB first.

14. (Currently Amended) The process of Claim 1, further comprising the step of:

resolving ~~reordering~~ frequency collisions in said list of reordered words when a user-defined word and a word from said ~~linguistic database~~ LDB have equal reordering frequency counts by ordering ~~the~~ said user-defined word first.

15. (Currently Amended) ~~The process of Claim 1,~~ A process for reordering items in a database to be retrieved for display to a user, comprising the steps of:

ordering a plurality of words stored in a linguistic database (LDB) according to a predefined linguistics frequency of use model;

associating each word stored in said LDB with an object number;

accepting user input from a keyboard;

retrieving any words from said LDB that match said user's input;

displaying a list of said retrieved words as ordered in said LDB;

if said list does not include a word, accepting said word as defined by said user;

storing said user-defined word in a user database (UDB);

assigning a frequency count to said user-defined word;

storing said frequency count in said UDB;

enabling said user to select a word from said displayed list;

assigning said frequency count to every selected word;

storing said frequency count of said selected word and said word's corresponding object number in said UDB;

accepting a subsequent user input;

retrieving any words from said LDB and any user-defined words from said UDB that match said user's subsequent input;

if more than one word matches said user's subsequent input, dynamically reordering any matching words in said LDB as ordered by said LDB and any matching words in said UDB according to said frequency count; and

displaying a list of said reordered matching words if more than one word matches;

wherein words selected by ~~the~~ said user that do not have a possibility of a collision with other words are not assigned ~~a~~ said ~~reordering~~ frequency count.

16. (Currently amended) An apparatus for reordering items in a database to be retrieved for display to a user, comprising:

a module for accepting user input from a keyboard;

a linguistic database (LDB) containing a plurality of words ordered according to a linguistic frequency of use model;

a module for displaying to said user a list of any words in said LDB and any user-defined words in a user database (UDB) that match said user input, said words retrieved from said LDB and from said UDB;

~~a user database (UDB) separate from the LDB that stores~~ said UDB for storing any user-defined words entered by said ~~that the~~ user, a frequency count for said user-defined words and any word selected by said user from said list of any words that match said user input, and an object number corresponding to said frequency count for any word selected by said user ~~enters into the module and that includes a reorder database (RDB) that stores database object numbers for words contained in the LDB~~;

a module for ~~dynamically~~ retrieving from ~~the~~ said LDB and from ~~the~~ said UDB a list of any words that match ~~include the sequence of letters formed by the~~ said user's input, said words dynamically reordered according to said frequency count; and

~~a module for displaying to the user a list of retrieved words, wherein the order of words determined by the RDB are displayed before the order of words determined by the LDB;~~

~~enabling the user to select a word from the displayed list; and~~

a module for assigning ~~a dynamic reordering~~ said frequency count to any words selected by ~~the~~ said user and inserting ~~the~~ said selected words' ~~assigned reordering~~ frequency counts into said ~~RDB~~ UDB.

17.-18. (Canceled)

19. (Currently Amended) The apparatus of Claim 16, wherein said frequency count is assigned to said selected word if said selected word is in a non first order position in said list and is selected for a first time ~~assigning module inserts into said reorder database UDB a non first ordered word from said list if the user selects the non first ordered word for the first time, and inserts into said reorder database the nonselected first ordered word if it does not already exist in said reorder database~~.

20. (Currently Amended) The apparatus of Claim 19, wherein if ~~the~~ said non first ordered word is selected by ~~the~~ said user a predetermined number of times then ~~the~~ said non first ordered word is displayed in a ~~assigned a higher frequency count than the~~ first ordered ~~word~~ position in said list.

21. (Currently Amended) The apparatus of Claim 19, wherein all non first ordered words that are assigned said frequency count ~~entered into said reorder database~~ are initially assigned equal ~~reordering~~ frequency counts.

22. (Currently Amended) The apparatus of Claim 16, wherein ~~a~~ said word's ~~reordering~~ frequency count is increased each time ~~the~~ said user selects ~~the~~ said word.

23. (Currently Amended) The apparatus of Claim 16, wherein if ~~the~~ said user selects from said list a word ~~which~~ that is below a second ordered position then said assigning module assigns a ~~value to the word's reordering~~ frequency count that places ~~the~~ said word in ~~the~~ said second ordered position in said list.

24. (Currently Amended) ~~The apparatus of Claim 16, further comprising:~~An apparatus for reordering items in a database to be retrieved for display to a user, comprising:

a module for accepting user input from a keyboard;

a linguistic database (LDB) containing a plurality of words ordered according to a linguistic frequency of use model;

a module for displaying to said user a list of any words in said LDB and any user-defined words in a user database (UDB) that match said user input, said words retrieved from said LDB and from said UDB;

said UDB for storing any user-defined words entered by said user, a frequency count for said user-defined words and any word selected by said user from said list of any words that match said user input, and an object number corresponding to said frequency count for any word selected by said user;

a module for retrieving from said LDB and from said UDB a list of any words that match said user's input, said words dynamically reordered according to said frequency count;

a module for assigning said frequency count to words selected by said user and inserting said selected words' frequency count into said UDB; and

a module for periodically aging ~~the reordering~~ said frequency counts in said reorder database by reducing ~~the reordering~~ said frequency counts by a predetermined factor

25. (Currently Amended) The apparatus of Claim 16, further comprising:

a module for periodically checking for free space in said ~~reorder database~~ UDB and, ~~if the~~ said free space is less than a predetermined amount, then removing from said UDB said ~~reorder database~~ frequency counts and corresponding object numbers or corresponding user-defined words for words that have ~~reordering~~ frequency counts below a predetermined threshold.

26. (Currently Amended) The apparatus of Claim 25, wherein said removing module removes user-defined words having ~~reordering~~ frequency counts below ~~the~~ said predetermined threshold after removing other words having ~~reordering~~ frequency counts below ~~the~~ said predetermined threshold.

27. (Currently Amended) The apparatus of Claim 16, further comprising:

a module for resolving ~~reordering~~ frequency collisions when two words in said list have equal ~~reordering~~ frequency counts by ordering ~~the~~ said most recently selected ~~of the two~~ words first.

28. (Currently Amended) The apparatus of Claim 16, further comprising:

a module for resolving ~~reordering~~ frequency collisions when two words in said list have equal ~~reordering~~ frequency counts by ordering ~~the~~ said word having a higher initial ordering in said ~~linguistic database~~ LDB first.

29. (Currently Amended) The apparatus of Claim 16, further comprising:

a module for resolving ~~reordering~~ frequency collisions in said list when a said user-defined word and a said word from said ~~linguistic database~~ LDB have equal ~~reordering~~ frequency counts by ordering ~~the~~ said user-defined word first.

30. (Currently Amended) ~~The apparatus of Claim 16, An apparatus for reordering items in a database to be retrieved for display to a user, comprising:~~ An apparatus for reordering items in a database to be retrieved for display to a user, comprising:

a module for accepting user input from a keyboard;

a linguistic database (LDB) containing a plurality of words ordered according to a linguistic frequency of use model;

a module for displaying to said user a list of any words in said LDB and any user-defined words in a user database (UDB) that match said user input, said words retrieved from said LDB and from said UDB;

said UDB for storing any user-defined words entered by said user, a frequency count for said user-defined words and any word selected by said user from said list of any words that match said user input, and an object number corresponding to said frequency count for any word selected by said user;

a module for retrieving from said LDB and from said UDB a list of any words that match said user's input, said words dynamically reordered according to said frequency count; and

a module for assigning said frequency count to words selected by said user and inserting said selected words' frequency count into said UDB;

wherein said words selected by ~~the~~ said user that do not have a possibility of a collision with other words are not assigned a said ~~reordering~~ frequency count.

31. (Currently Amended) The process of Claim 1 wherein when a said word from ~~the~~ said LDB is selected for ~~the~~ a first time, said step of assigning said frequency count generates said frequency count from ~~uses the~~ said word's position in said LDB as a function of said predefined liquistics frequency of use model ~~order in the LDB as an initial dynamic reordering frequency count~~.

32.-33. (Canceled)

34. (Currently Amended) The process of Claim 31, wherein said ~~all words specifically entered by the~~ user-defined words are initially assigned equal ~~reordering~~ frequency counts ~~by~~ during said assigning step.

35.-44. (Canceled)

45. (Currently Amended) The apparatus of Claim 16 wherein ~~when a word from the LDB is selected for the first time,~~ said assigning module assigns said frequency count for said word from said ~~for assigning uses the word's frequency of use order in the~~ LDB based on said predefined linquistics frequency of use model ~~as an initial dynamic reordering frequency count~~.

46.-47. (Canceled)

48. (Currently Amended) The apparatus of Claim 45, wherein all saiduser-defined words ~~specifically entered by the user~~ are initially assigned equal ~~reordering~~ frequency counts by said assigning module.

49.-60. (Canceled)